Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Administração de Base de Dados Oracle e Desenvolvimento de Metodologias de Suporte

por

Nuno de Jesus Salvador

Mestrado em Engenharia Informática

2008

Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Administração de Base de Dados Oracle e Desenvolvimento de Metodologias de Suporte

Nuno de Jesus Salvador

PROJECTO

Projecto orientado pelo Prof. Dr. Hans Peter Reiser
e co-orientado por Nuno Miguel de Sousa Maria

Mestrado em Engenharia Informática

2008

Declaração

Nuno de Jesus Salvador, aluno nº 31569 da Faculdade de Ciências da Universidade de Lisboa, declara ceder os seus direitos de cópia sobre o seu Relatório de Projecto em Engenharia Informática, intitulado "Administração de Base de Dados Oracle e Desenvolvimento de Metodologias de Suporte", realizado no ano lectivo de 2007/2008 à Faculdade de Ciências da Universidade de Lisboa para o efeito de arquivo e consulta nas suas bibliotecas e publicação do mesmo em formato electrónico na Internet.

FCUL, 30 de Setembro de 2008

Nuno Miguel de Sousa Maria, supervisor do projecto de *Nuno de Jesus Salvador*, aluno da Faculdade de Ciências da Universidade de Lisboa, declara concordar com a divulgação do Relatório do Projecto em Engenharia Informática, intitulado "Administração de Base de Dados Oracle e Desenvolvimento de Metodologias de Suporte".

Lisboa, 30 de Setembro de 2008

Resumo

Este documento descreve o projecto realizado no âmbito da disciplina Projecto em Engenharia Informática do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa.

Actualmente, para grande parte das organizações, a sua competência técnica reside no conhecimento adquirido pelos seus técnicos. A rotação dos recursos humanos acaba por ameaçar as valências e capacidade das empresas. Torna-se assim necessário consolidar o conhecimento, partilhando-o e armazenando-o, para que a empresa se mantenha competitiva e possa apresentar uma solução rápida e compatível com as exigências dos seus clientes.

Actualmente, no mercado, existem diversas soluções e plataformas de *Service Desk* e Bases de Conhecimento, quer comerciais quer *"open-source"*, que endereçam esta problemática. No entanto, apesar de algumas destas serem bastante evoluídas, não é vulgar encontrar um software que permita a gestão dos pedidos de intervenção bem como o armazenamento, de forma estruturada, das acções e soluções adoptadas para cada incidente.

Este projecto teve como objectivo o desenvolvimento de uma aplicação que permita por um lado, gerir o trabalho desenvolvido ao longo de uma intervenção, através do registo das actividades que vão sendo realizados e, por outro, com este registo, criar uma Base de Conhecimento que relacione a informação de forma inteligente, evite duplicações, e que possibilite uma pesquisa fácil com resultados relevantes.

O protótipo resultado do processo de desenvolvimento, a KD, que integra duas aplicações, Mantis e MediaWiki, com outros componentes desenvolvidos à medida, permitiu aumentar a eficácia na resolução de incidentes no ambiente onde foi instalado, havendo ainda, no entanto, margem para progressão e melhorias.

PALAVRAS-CHAVE:
Service Desk, Base de Conhecimento, Mantis, MediaWiki, Oracle

Abstract

This document describes the project implemented for the course Project in Informatics Engineering of the Masters in Informatics Engineering of the Faculty of Sciences at Lisbon University.

Company technical skills are often based upon the professional experience of its staff. Many organizations rely today on their human resources knowledge about its projects and systems. Job rotation is a real threat against these companies's technical expertise and capability. It becomes necessary to assure that critical knowledge is stored and shared between company staff, so that customer requests and requirements are promptly satisfied.

Available commercial and open-source solutions for service desk and knowledge base purposes are extensive. However, despite the many advanced features found in several solutions, it is not common to find integrated software solutions that allow management of the service requests and, in a non-redundant structured way, store the knowledge needed to address similar situations in the future. KD addresses this problem.

KD is a software solution built integrating Mantis, MediaWiki with other custom developments and components. It allows standard task management and execution needed to address service requests, providing in the same integrated application, tools to create, on the fly, an intelligent and non-redundant knowledge base.

The prototype that was built, allowed increase on resolution efficiency of service requests through the use of the knowledge base built using KD features. Nevertheless, it can be significantly improved.

KEYWORDS:
Service Desk, Knowledge Base, Mantis, MediaWiki

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Esta dissertação apresenta e descreve o projecto realizado no âmbito da disciplina do PEI, leccionada na FCUL no ano lectivo de 2007/2008. Este projecto foi realizado na empresa *Práxia - Sistemas de Informação, SA* e centrou-se na administração de Sistemas Oracle e na plena integração de uma Base de Conhecimento com um software de registo de incidentes e intervenções, *Service Desk*.

1.1 Motivação

Actualmente, para grande parte das organizações, a sua competência técnica reside no conhecimento adquirido pelos seus técnicos. A rotação dos recursos humanos, própria de um mercado laboral cada vez mais competitivo, acaba por ameaçar as valências e capacidade das empresas. Torna-se assim necessário consolidar o conhecimento, partilhando-o e armazenando-o, para que a empresa se mantenha competitiva e possa apresentar uma solução rápida e compatível com as exigências dos seus clientes.

Actualmente, no mercado, existem diversas soluções e plataformas de *Service Desk* e Bases de Conhecimento, quer comerciais quer *"open-source"*. No entanto, apesar de algumas destas serem bastante evoluídas, não é vulgar encontrar um software que permita a gestão dos pedidos de intervenção bem como o armazenamento, de forma estruturada, das acções e soluções adoptadas para cada incidente.

As Bases de Conhecimento, que manipulam quantidades enormes de informação, acabam por tornar difícil a pesquisa de algo concreto sem que apareçam resultados inúteis e repetidos. E, se são limitadas, acabam por não possuir informação útil. Por outro lado, as aplicações de *Service Desk* são normalmente autónomas e não privilegiam a recolha de informação estruturada para a Base de Conhecimento. Estas são lacunas que foram explorados no desenvolvimento do projecto e que a "KD - Knowledge Desk"tenta endereçar.

1.2 Objectivos

Este projecto teve como objectivo o desenvolvimento de uma aplicação que permita por um lado, gerir o trabalho desenvolvido ao longo de uma intervenção, através do registo das actividades que vão sendo realizados e, por outro, com este registo, criar uma Base de Conhecimento que relacione a informação de forma inteligente, evite duplicações, e que possibilite uma pesquisa fácil com resultados relevantes.

Assim, através de uma interface *Web*, o KD deverá permitir:

- Introdução selectiva e segura de informação relevante para o negócio;
- Introdução organizada e rica que permita criar uma rede de informação útil;
- Pesquisa de informação envolvendo mecanismos de determinação e relevância avançada.

A solução desenvolvida que integra uma aplicação de *Service Desk*, o Mantis, com uma Base de Conhecimento, a MediaWiki, possui ainda funções avançadas de estruturação e arquitectura de informação que permitem relacionar informação de múltiplas formas.

1.3 Organização do documento

Neste documento, e depois desta introdução que resume os objectivos e o contexto do projecto, introduzem-se no capítulo 2 algumas referências a produtos existentes no mercado, bem como suas vantagens e desvantagens. No capítulo 3 descreve-se a metodologia usada no desenvolvimento do projecto. No capítulo 4 é apresentada a arquitectura do sistema e os seus componentes. No capítulo 5 é feita uma análise cuidada e crítica a todo o desenvolvimento realizado bem como os resultados obtidos e os testes efectuados. Finalmente no 6 são apresentadas as conclusões com direcções para trabalho futuro.

1.4 Integração Profissional

Em Setembro de 2007 iniciou-se o estágio nas instalações da *Práxia*. A *Práxia* é uma empresa de sistemas de informação que conta com especialistas certificados no desenvolvimento e suporte de soluções baseadas em tecnologias Java, Microsoft, Oracle e Outsystems.

Os primeiros dias na *Práxia* tiveram como objectivo dar a conhecer a estrutura da empresa, as competências de cada unidade orgânica, a sua organização, normas, métodos de trabalho e a sua política de qualidade.

Após este período de ambientação, teve início um período de auto-formação com conceitos associados às base de dados Oracle [22], abrangendo a arquitectura das soluções assentes em tecnologia Oracle.

Simultaneamente, e com a integração na equipa da Unidade Oracle que coincidiu com o início de um projecto, houve lugar à instalação da base de dados de suporte à aplicação Meta4 [20]. Foi também estudada e instalada esta aplicação e todas as suas componentes - *Servidor Aplicacional*, *Servidor Web* com *Web Container*. Após estas tarefas foram, pela primeira vez, analisados sistemas de *Service Desk*, seu funcionamento e potencial.

Depois desta fase, que se prolongou um pouco mais do que o esperado, seguiu-se o desenrolar do projecto proposto com a transferência do conhecimento acerca doutro projecto, já em curso, nas Oficinas Gerais de Material Aeronáutico (*OGMA*), de suporte aos sistemas operativos e base de dados Oracle e que serviu de infra-estrutura ao desenvolvimento da KD, ao longo dos últimos meses. Neste cliente foram ainda desenvolvidas actividades de manutenção do software Oracle e sistema operativo Linux. Ao longo deste período existiram, ainda, reuniões semanais/quinzenais, nas quais era avaliado o trabalho desenvolvido até então e aferido o planeamento do projecto.

Capítulo 2

Contexto e Enquadramento Tecnológico

2.1 Trabalho Relacionado e Outras Soluções

2.1.1 Aplicações de Base de Conhecimento

Uma base de conhecimento mundialmente conhecida é a Wikipédia [32]. É uma enciclopédia livre em que um interveniente pode acrescentar, modificar e/ou apagar artigos da forma que pretender. Esta é baseada no sistema WikiWeb [30], ou simplesmente Wiki.

O sistema WikiWeb permite a edição de documentos online, por uma comunidade, utilizando apenas um vulgar browser Web. Para tal, basta que seja criada uma conta através de um rápido registo no sistema, onde é pedido um nome e uma senha, passando o utilizador, logo de imediato, a ter acesso a novas opções como corrigir erros, completar ideias e inserir novas informações. Mas, o que por um lado pode ser uma vantagem, com o rápido crescimento dos artigos e informação associada, por outro torna-se uma desvantagem, pois nada impede um utilizador de remover artigos ou modificá-los com informações erradas. Outra desvantagem é o facto da introdução de artigos não ser controlada nem selectiva, havendo, por vezes, redundância na informação.

De seguida são apresentados dois exemplos que, no contexto da realização da Base de Conhecimento, foram avaliados [31].

MediaWiki

A Wikipédia usa o software desenvolvido pela MediaWiki [19]. Esta aplicação foi desenvolvida em PHP [10], sob uma licença GPL [13][1]. Para o armazenamento e

[1] *GNU [14] General Public license* é o nome dado a uma licença de suporte ao Software livre. Esta licença, que vai já na 3ª versão, tem como premissas, entre outras: A liberdade de executar o programa para qualquer propósito; Estudar e alterar um programa de acordo com as necessidades

tratamento da informação a MediaWiki recorre a uma base de dados MySQL [21], podendo, no entanto, serem usadas base de dados PostgreSQL [24] ou Oracle, para as quais já existem extensões. O que torna este software um dos utilizados nesta área é a sua escalabilidade, quantidade de plugins existentes e o facto de ser muito completo, permitindo, por exemplo, ver as diferenças entre revisões anteriores, discutir as revisões actuais, organizar os artigos por categorias e *namespaces*, entre outras.

DokuWiki

Outra aplicação muito usada para Base de Conhecimento, e que foi tida em conta na realização deste trabalho, é a DokuWiki [11]. Ao contrário da MediaWiki a DokuWiki assenta numa política de armazenamento ao nível de ficheiros ao invés de usar uma base de dados, facilitando assim a leitura dos artigos fora da wiki bem como a criação de textos estruturados. Também desenvolvido em PHP, e com licença GPLv2, este software caracteriza-se por ser muito mais leve e mais fácil de instalar, configurar e manter, do que a MediaWiki, não perdendo, no entanto, características como a fácil personalização e adaptação, muito devido ao sistema de plugins que possui.

2.1.2 Aplicações de Service Desk

Uma aplicação de *Service Desk* é um sistema de registo de incidentes e pedidos de apoio para os utilizadores de um serviço ou sistema, actuando normalmente como uma primeira linha de contacto e uma forma de fazer registo de incidentes em actividades de suporte ao sistema de informação. Um *Service Desk* regista incidentes ou pedidos através de múltiplos canais como chamadas telefónicas ou e-mails enviados pelos utilizadores reportando problemas a serem resolvidos. Os problemas são registados na aplicação, sendo-lhe normalmente atribuída uma classificação e um nível de prioridade. Através da aplicação é possível gerir o ciclo de vida do pedido, escalar o problema, notificar o utilizador que a resolução está em curso e solicitar informação adicional ao utilizar. A grande vantagem de todo o procedimento, para além de uma maior eficiência e rapidez na resolução do pedido, é o registo deste, caso seja preciso mais tarde consultar o que realmente se passou.

Algumas empresas como a AdventNet [2], a Ilient [9] ou a CA [5] desenvolvem aplicações de *Service Desk* de grande qualidade, em termos de design, funcionalidades e metodologias, partindo já de princípios ITIL[2]. No entanto, estas aplicações

do utilizador; Redistribuir o programa gratuitamente; disponibilizar o código-fonte e todas as alterações feitas.

[2] *ITIL - Information Technology Infrastructure Library* é o nome dado a um conjunto de bibliotecas consistentes e compreensivas de boas práticas/maneiras em gestão de serviços em Tecnologias de Informação.

não permitem a recolha de informação e a sua pesquisa por incidentes antigos torna-se difícil e cansativa.

Mantis

O Mantis [18] é uma aplicação *Web*, desenvolvida em PHP e difundida sob licença GPL. É conhecida por ser uma aplicação de fácil instalação, utilização e aprendizagem. O Mantis é uma ferramenta mais orientada ao controlo e gestão de problemas de uma aplicação (*Bug Tracker* [4]). O Mantis permite aos programadores manter um registo dos erros reportados do sistema desenvolvido/a desenvolver. As principais características do Mantis são: baixo custo ao nível do hardware, administração, suporte e integração com vários sistemas operativos; a possibilidade de gerar gráficos de estatísticas; possibilidade de ter vários projectos agrupados por categorias; campos modificáveis para cada projecto; possibilidade de criar um *workflow* com atribuição de responsabilidades, entre outras.

Para além disto esta aplicação conta já com uma API para uma rápida integração com algumas das base de conhecimento mais vulgares (como a MediaWiki e a Dokuwiki) criando em cada incidente um *link* para a Base de Conhecimento, com o número do incidente no endereço. O Mantis usa uma base de dados MySQL como suporte, podendo no entanto, ser utilizado com bases de dados MS SQL [26], PostgreSQL ou mesmo Oracle (esta última ainda a nível experimental).

OpenView Service Desk

O HP [6] OpenView [29] é um conjunto de produtos de gestão de sistemas e redes em grande escala que inclui uma grande variedade de módulos/plugins tanto da HP como de outras empresas. Uma das aplicações incluídas nesta suite é uma aplicação de *Service Desk*. Esta aplicação é executada numa *Java Virtual Machine* da Microsoft, e é mais complexa e pesada que o Mantis, não possuindo uma interface *Web*, sendo necessário, por isso, instalar uma aplicação em cada cliente. Com o HP OpenView Service Desk é possível monitorizar, notificar e analisar todos os pedidos feitos pelos utilizadores bem como obter gráficos de estatísticas dos pedidos e dos técnicos. Foi construída sobre princípios ITIL seguindo as *Best Practices* da indústria das Tecnologias de Informação.

CA Service Desk

Outra aplicação de *Service Desk* testada neste processo foi a solução oferecida pela CA [8]. Uma aplicação *Web* onde as grandes características são as divisões por projectos, sub-projectos e categorias, sendo possível modificar cada campo de acordo com o projecto. É também possível ter vários tipos de utilizadores no sistema sendo

permitido aos técnicos adicionar notas/comentários transparentes para os outros utilizadores.

2.2 Tecnologia

Apresenta-se de seguida uma breve descrição das tecnologias, onde assentam algumas das soluções atrás descritas, que servirão de base ao desenvolvimento da KD e que lhe conferem, por inerência, algumas das suas características.

2.2.1 SQL

O SQL (Structured Query Language) [15] é uma linguagem de pesquisa normalizada que permite a recolha e gestão da informação presente num sistema de base de dados.

2.2.2 PHP

O PHP [10] é uma linguagem de programação que, quando combinada com um servidor *Web*, pode ser usada para produzir páginas *Web* dinâmicas. É uma das linguagens mais usadas para este tipo de função e deve a sua popularidade ao facto de poder ser usada com um vasto número de sistemas de gestão de bases de dados (incluindo Oracle), de estar disponível para muitos sistemas operativos, de poder ser executada na maioria dos servidores Web e ser uma linguagem relativamente de fácil aprendizagem. Esta é uma linguagem *server-side*, como tal executa do lado do servidor sendo que apenas é apresentado ao utilizador o *output* gerado.

2.2.3 XML

O XML (eXtensible Markup Language) [25] é uma metalinguagem independente da plataforma lógica e física. Permite descrever informação de uma forma bem definida. Um documento XML tem uma estrutura composta por elementos que podem, ou não, ter atributos. Cada elemento descreve uma parte da informação que está contida no documento XML. A estrutura de um documento XML obedece a um conjunto de regras estritas e só é válido se estas regras forem cumpridas. Esta característica garante que documentos XML provenientes de fontes distintas, possam ser interpretados da mesma forma permitindo a interoperabilidade da informação através de sistemas heterogéneos através da sua normalização.

2.2.4 JavaScript

O JavaScript [12] é uma linguagem interpretada. Foi desenvolvida para deslocar a execução de código para o cliente na obtenção de páginas HTML através da interpretação dinâmica das instruções presentes na página, possibilitando a construção

de aplicações *Web* interactivas através da alteração da estrutura das páginas com a interacção com o *browser*.

2.2.5 AJAX

O AJAX (Asynchronous JavaScript And XML) [33] é uma tecnologia baseada na utilização de JavaScript e XML para a troca de informação assíncrona entre um cliente (*Web browser*) e um servidor (*Web server*), que possui vantagens sobre o modelo tradicional de cliente/servidor na *Web*. Face ao modelo tradicional, o AJAX veio melhorar a eficiência da troca de informação entre cliente e servidor, permitindo a comunicação assíncrona entre eles. Uma das grandes vantagens é o facto do cliente não ter de ficar à espera da resposta do servidor para continuar a desempenhar as suas funções, prosseguindo imediatamente a seguir ao pedido e disponibilizando a informação ao utilizador à medida que esta é recebida do servidor. Esta tecnologia permite alterar de forma dinâmica uma página sem ter que a carregar toda por completo.

2.2.6 CSS

As Cascading Style Sheets [16] definem a apresentação e aspecto dos elementos de uma página HTML, permitindo simplificar e uniformizar a apresentação das páginas HTML, contribuindo para um aspecto mais coerente e limpo da aplicação.

2.2.7 Oracle Intermedia Text

O *Oracle Intermedia Text* [28] é uma ferramenta da Oracle disponibilizada em conjunto com a Base de Dados. Esta ferramenta permite a indexação de conteúdos como texto, documentos, imagens, áudio e vídeo, através de uma estrutura de índice invertido[3], o que leva a uma gestão eficiente e rápida destes através de simples queries SQL ou packages PL/SQL.

[3]O *Oracle Intermedia Text* indexa os itens através de palavras-chave, onde a cada palavra fica associado em que artigos ocorre.

Capítulo 3

Metodologia

Nas secções seguintes é apresentado o processo de desenvolvimento seguido e o planeamento estabelecido para o desenvolvimento da KD.

3.1 Processo de Desenvolvimento

Para o desenvolvimento do projecto foi seguida uma metodologia em espiral [3]. Pretendeu-se com esta opção basear o desenvolvimento numa metodologia rápida e ágil onde o objectivo foi minimizar o risco do desenvolvimento do software através de iterações.

O modelo em espiral é caracterizada por combinar, em etapas sucessivas, as fases de desenho e implementação no desenvolvimento de um projecto de *software*. Devido a esta combinação, este modelo apresenta algumas vantagens importantes, das quais se destacam as seguintes:

- As estimativas (de custo, duração do projecto, etc.) tornam-se mais realistas e previsíveis à medida que o trabalho progride, pois as principais dificuldades técnicas são detectadas no início do desenvolvimento do projecto;
- Permite uma melhor adaptação às alterações no desenho e arquitectura do projecto em curso, espelhando rapidamente estas alterações na fase de implementação;
- Os programadores podem antecipar problemas mais cedo nas tarefas de implementação e desenvolvimento, em vez de ficarem presos nas fases de análise e desenho.

Na figura 3.1 é apresentado um diagrama que ilustra as várias fases no desenvolvimento da KD com detalhe à metodologia iterativa aplicada.

O projecto iniciou-se com uma fase de adaptação que incluiu a integração na empresa e correspondeu à introdução, formação e exploração das tecnologias Oracle,

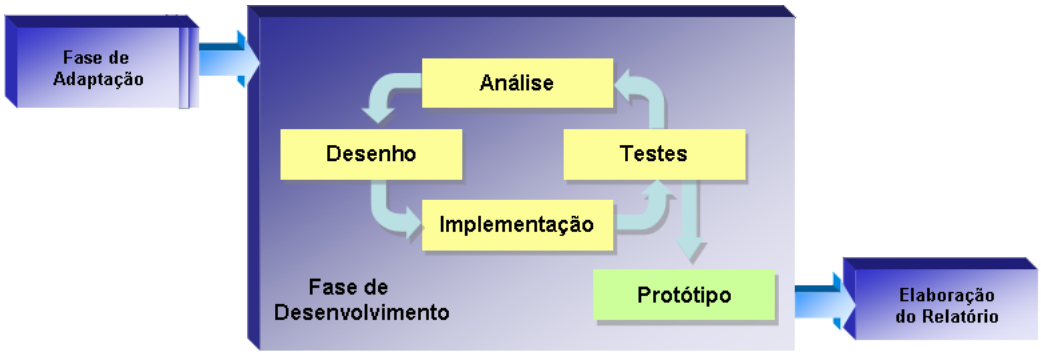


Figura 3.1: Fases de desenvolvimento do projecto

linux, e servidores aplicacionais, bem como a integração num projecto em curso onde eram patentes as necessidades tanto de uma Base de Conhecimento como de um *Service Desk*.

Na fase de desenvolvimento, as actividades que sofreram várias iterações ao longo do ciclo do desenvolvimento são representadas pelas caixas análise, desenho, implementação e testes. Para uma melhor compreensão apresenta-se, de seguida, uma breve descrição das actividades de cada uma das etapas.

- **Análise**: esta etapa correspondeu ao levantamento e análise de requisitos bem como o estudo de tecnologias, meios e ferramentas a utilizar no desenvolvimento do projecto. Foram identificadas possíveis aplicações de *Service Desk* e base de conhecimento *"open-source"*. Foram instaladas e testadas as que melhor se adequaram aos objectivos do projecto;

- **Desenho da Solução**: nesta etapa foram estudadas as aplicações escolhidas, bem como a melhor forma de as integrar e criar uma estrutura que permitisse concretizar uma rede de informação rica e selectiva;

- **Desenvolvimento**: foi feita a implementação da aplicação levando a cabo o objectivo proposto, tendo em conta todo o estudo, análise e desenho realizados em cada iteração;

- **Avaliação e Testes**: nesta etapa foi feita uma avaliação do desenvolvimento. Foram avaliadas as escolhas realizadas na fase de análise e se o desenho para a rede de informação sustentava os objectivos do projecto, descortinando melhorias que podiam vir a ser necessárias.

- **Protótipo**: Em cada iteração era objectivo a produção de um protótipo que implementasse as funcionalidades concretizadas nas várias iterações realizadas.

Foram executadas 4 iterações até à conclusão dos desenvolvimentos. Na próxima secção são descritos os detalhes relativos ao seu planeamento e execução.

3.2 Planeamento

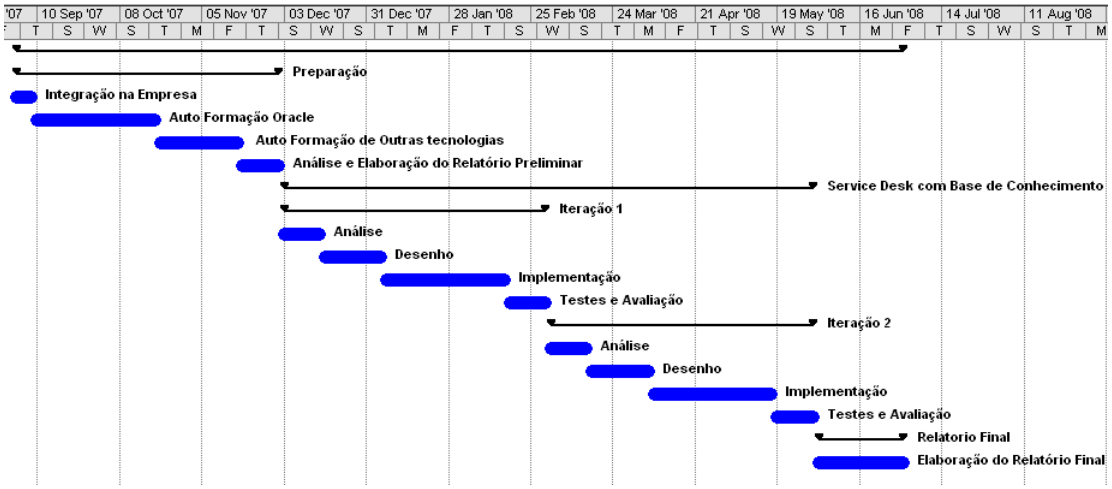


Figura 3.2: Planeamento Inicial

O planeamento das actividades ocorreu no início do projecto e previa uma duração de 9 meses e 2 iterações, como apresentado na figura 3.2.

No entanto, devido ao prolongamento de algumas das etapas da fase de Adaptação, originalmente prevista para durar cerca de 3 meses, mas que foi excedido em cerca de 2 meses adicionais, foi necessário reformular o planeamento do projecto. Esta derrapagem foi consequência de um alargamento do âmbito das funções atribuídas pela empresa a que foi necessário dar resposta e que envolveu o domínio de tecnologias como o *Software* aplicacional *Meta4*, e os componentes *Oracle SOA framework*, *Oracle Universal Content Management* e *Oracle Business Intelligence*. A tabela 3.1, detalha as tarefas executadas na fase de Adaptação.

Tarefa	Duração	Início	Fim
Adaptação	**100 dias**	**03/Set/2007**	**18/Jan/2008**
Integração na Empresa	05 dias	03/Set/2007	07/Set/2007
Auto Formação Oracle	30 dias	10/Set/2007	19/Out/2007
Auto Formação em Outras Tecnologias	20 dias	22/Out/2007	16/Nov/2007
Análise e Elaboração do Relatório Preliminar	10 dias	19/Nov/2007	30/Nov/2007
Auto Formação Meta4 / SOA / UCM / BI	35 dias	03/Dez/2007	18/Jan/2008
Adaptação - Fim		**20/Jan/2008**	**20/Jan/2008**

Tabela 3.1: Fase de adaptação: Tarefas e duração final

Sendo assim, as duas iterações inicialmente planeadas, de 3 meses cada, foram revistas. Em função desta revisão a abordagem seguida, para além de diminuir

o âmbito e duração de cada uma das fases, foi aumentar o número de iterações, reduzindo os objectivos em cada uma das iterações e minimizando o risco do desenvolvimento.

A primeira iteração coincidiu com a selecção, instalação, configuração e desenvolvimento da componente de *Service Desk* escolhida para o projecto. A segunda iteração, que decorreu desde meados de Março até ao final de Abril, coincidiu com a selecção, instalação, configuração e desenvolvimento da componente de Base de Conhecimento escolhida. Tanto na primeira, como na segunda iteração, os grandes objectivos foram verificar que as aplicações escolhidas eram adequadas para satisfazer os requisitos do projecto. Foi ainda alcançada nesta iteração, a migração das bases de dados dos componentes seleccionados inicialmente de MySQL para Oracle. Na segunda iteração antes de se chegar ao compromisso com a base de conhecimento escolhida, ainda foi testado uma outra aplicação, que à partida parecia ser a melhor opção, o que não se veio a verificar, depois de alguns testes e planos de integração desta com a base de dados Oracle.

Depois dos testes que garantiram o correcto funcionamento destas duas aplicações, foram iniciados os trabalhos de integração e consolidação com o componente *Knowledge Desk Component*, que visava garantir a plena integração das duas aplicações. Este componente foi desenvolvido na 3ª e 4ª iterações com objectivos de pequena dimensão que foram sendo alcançados, para garantir a estabilidade e fiabilidade do produto final, bem como a sua instalação num ambiente de produção. Os objectivos, na 3ª iteração, passaram por permitir adicionar incidentes criados no *Service Desk* directamente na Base de Conhecimento e permitir a pesquisa de artigos na Base de Conhecimento através da aplicação de *Service Desk* com o uso da tecnologia Oracle Text. Na 4ª iteração foi optimizada a aplicação de modo a permitir a escolha do nome do artigo a criar na Base de Conhecimento, através do *Service Desk*, a criação automática de *namespaces* na Base de Conhecimento e a criação de uma melhor ligação entre os incidentes do *Service Desk* e os artigos criados na Base de Conhecimento.

De seguida é apresentado na tabela 3.2 o detalhe da execução do planeamento do projecto.

Tarefa	**Duração**	**Início**	**Fim**
Adaptação	**4.5 meses**	**03/Set/2007**	**18/Jan/2008**
1ª Iteração	**5 semanas**	**21/Jan/2008**	**22/Fev/2008**
Análise	1 semana		
Desenho	1 semana		
Implementação	2 semanas		
Testes	1 semana		
2ª Iteração	**5 semanas**	**25/Fev/2008**	**28/Mar/2008**
Análise	1 semana		
Desenho	1 semana		
Implementação	2 semanas		
Testes	1 semana		
3ª Iteração	**10 semanas**	**1/Abr/2008**	**6/Jun/2008**
Análise	1 semana		
Desenho	2 semanas		
Implementação	5 semanas		
Testes	2 semanas		
4ª Iteração	**5 semanas**	**17/Jun/2008**	**25/Jul/2008**
Análise	1 semana		
Desenho	1 semana		
Implementação	1 semana		
Testes	2 semanas		

Tabela 3.2: Fases do Projecto: Tarefas, calendarização e duração final

Capítulo 4

Arquitectura Do Sistema

A arquitectura da KD foi concebida tendo em conta a integração de uma aplicação de *Service Desk* com uma Base de Conhecimento. O desenvolvimento de raiz de um sistema para *Service Desk* e outro para suportar uma Base de Conhecimento não faz sentido, pois existem hoje em dia soluções de elevada qualidade disponíveis. Apesar de não serem encontradas com facilidade sistemas que integram estas duas funções, não deve ser menosprezado todo o esforço empregue no seu desenvolvimento. Por outro lado, repetir o desenvolvimento destes dois componentes alargaria o âmbito deste projecto de forma incomportável. Assim, e além do objectivo de conjugar ferramentas fiáveis e de qualidade, foram ainda objectivos no desenho da arquitectura do sistema:

- **Portabilidade e Interface Web**: Pretende-se que a aplicação possa ser instalada em ambientes heterogéneos, isto é, independente da tecnologia e Sistema Operativo. Pretende-se também que qualquer utilizador possa aceder à aplicação através do seu computador de trabalho, de maneira rápida e cómoda, sem ser necessário a instalação de aplicações adicionais em cada nó cliente;
- **Escalabilidade**: a KD deverá suportar o aumento do número de utilizadores bem como o aumento de pedidos e artigos sem pôr em causa o seu bom funcionamento nem o seu desempenho. Pretende-se também que, com o aumento gradual dos artigos na KD, esta não perca a sua estrutura nem a organização selectiva dos artigos, continuando a facilitar a pesquisa destes;
- **Baixo Custo**: a KD deverá ser suportada pelo mínimo custo, dando-se primazia a tecnologias *"open-source"* e aplicações já existentes no mercado, baixando, assim, o custo a nível de licenciamento e não consumindo recursos a conceber novas formas de produzir o que já existe.

Com base nestas condicionantes, foram avaliadas as aplicações presentes no mercado, tanto para *Service Desk* como para Base de Conhecimento, que melhor se enquadravam e foi desenhada a arquitectura pretendida para a KD.

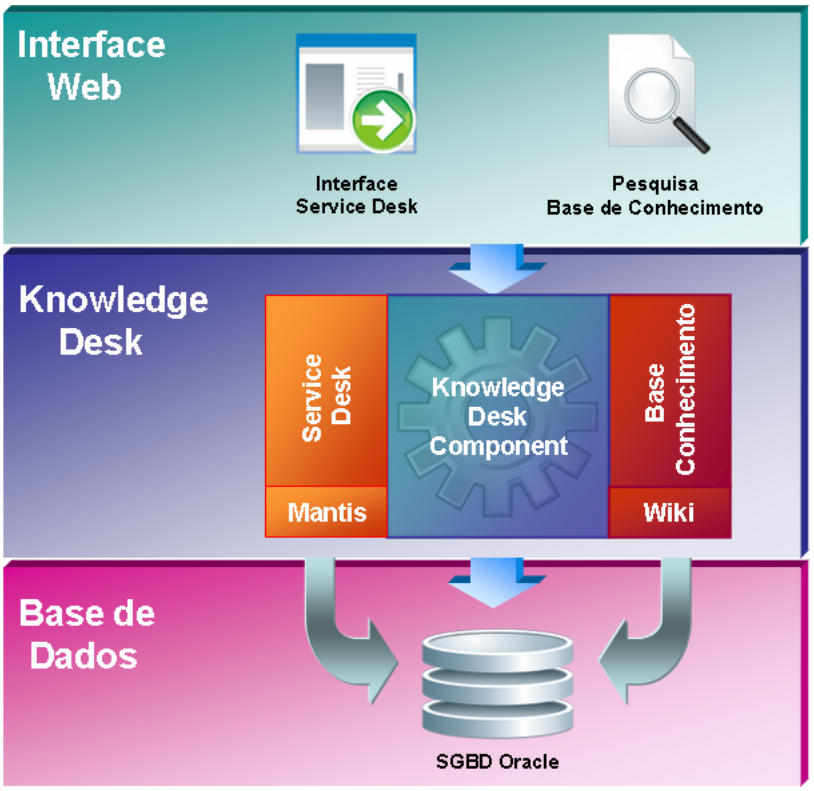


Figura 4.1: Arquitectura da KD

A figura 4.1 apresenta o diagrama da arquitectura desenhada da KD. A arquitectura da KD é composta por três grandes camadas:

1. **Base de Dados**: parte crítica da solução, pois é este componente que sustenta toda a informação introduzida pelos utilizadores e suporta todas as pesquisas.

2. ***Knowledge Desk***: Camada aplicacional que integra três sub-componentes:

   - **Motor do *Service Desk***: permite realizar as operações de registo, edição, consulta e administração de incidentes e ocorrências. Este componente é concretizado pelo software Mantis;
   - **Motor da Base de Conhecimento**: permite executar opções de pesquisa avançadas e inserção e manipulação explícita de conteúdos. Este componente é concretizado pelo software MediaWiki;
   - **Knowledge Desk Component**: concretiza as ligações entre a Base de Conhecimento, o *Service Desk* e a base de dados. Aqui são definidas todas as regras, de maneira a que toda a aplicação funcione de forma consolidada e eficaz. Este sub-componente torna possível acrescentar informação relevante e selectiva na Base de Conhecimento, permitindo, ao mesmo tempo, uma pesquisa eficaz nesta.

3. **Interface *Web***: concentra toda a interface com o utilizador através de tecnologia *Web* utilizada pela aplicação de *Service Desk* e Base de Conhecimento. Este componente é composto essencialmente por duas grandes módulos:

   - **Interface *Service Desk***: permite o correcto funcionamento da aplicação de registo de incidentes e pedidos de apoio;
   - **Pesquisa na Base de Conhecimento**: permite executar pesquisas avançadas e adicionar explicitamente informação na base de conhecimento.

Nas próximas secções é descrito com maior detalhe cada um dos componentes da arquitectura.

4.1 Base de Dados

Este componente armazena toda a informação introduzida na aplicação sendo usado o SGBD Oracle 10g. Como as aplicações de *Service Desk* e Base de Conhecimento a usar assentam em software *"open-source"*, as estruturas de dados são provenientes das aplicações escolhidas. As únicas alterações efectuadas resultam da necessidade de adicionar as tabelas de suporte ao Knowledge Desk Component, bem como garantir a plena integração das aplicações com base de dados Oracle, dado que a grande maioria das aplicações *"open-source"* estão preparadas para o uso de base de dados MySQL.

A opção por retirar o suporte da base de dados MySQL das aplicações de *Service Desk* e Base de Conhecimento e substituí-la pelo SGBD Oracle, baseou-se em dois grandes factores:

- Integração com base de dados já em uso nos clientes alvo da KD;
- Disponibilização nativa e integrada no Oracle de funções de IR - *Information retrieval [17]* - através do componente *Oracle Text*.

Na figura 4.2 é apresentada a entidade, `Mantis_Wiki_Table`, adicionada aos modelos de dados já existentes das aplicações de *Service Desk* e Base de Conhecimento bem como a integração desta com as restantes tabelas. Esta tabela serve de elo de ligação às duas aplicações empregues sendo que a cada identificador de um incidente ou ocorrência (`MANTIS_BUG_ID`) fica associado ao artigo correspondente na Base de Conhecimento através do seu *namespace* [1] (`WIKI_PAGE_NAMESPACE`) e título (`WIKI_PAGE_TITLE`).

[1] Namespace ou Espaço de Nomes é um objecto abstracto, que contém outros objectos e que fornece contexto para os itens que armazena e oferece desambiguação para itens que possuem o mesmo nome, mas residem em espaços de nomes diferentes.

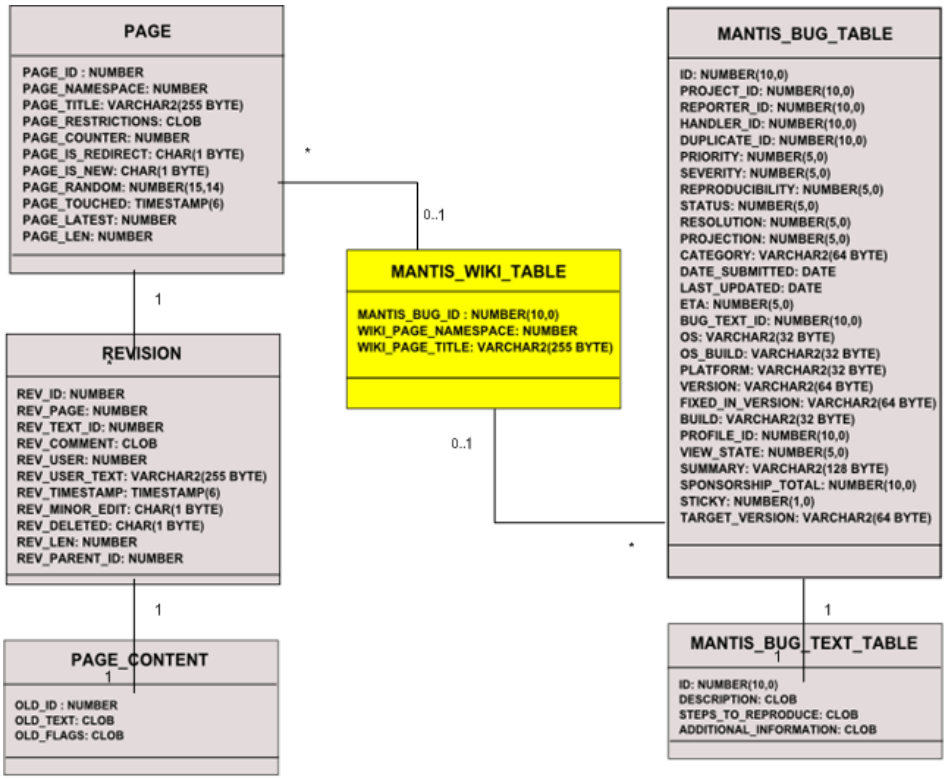


Figura 4.2: Tabelas Adicionais da base de dados

A comunicação entre esta camada com as outras camadas da aplicação é assegurada pela linguagem SQL, uma vez que é esta que permite que os dados recolhidos sejam introduzidos e consultados na base de dados.

4.2 Knowledge Desk

Esta camada é a responsável pela operação do KD. É composta por três grandes componentes:

Motor do *Service Desk*

Este componente é responsável pela manipulação de todos os incidentes e ocorrências introduzidos. É concretizado através do software Mantis. O Mantis tem uma arquitectura baseada nos componentes apresentados no diagrama da figura 4.3.

Através da sua interface, desenvolvida em PHP, é possível administrar, inserir e editar incidentes e ocorrências bem como os projectos e utilizadores envolvidos. Através do Mantis é também possível receber alertas no e-mail de actualizações a um incidente.

Figura 4.3: *Service Desk*: Arquitectura e funcionalidades

Motor da Base de Conhecimento

Este componente é responsável pela manipulação da Base de Conhecimento do KD. É concretizado com software MediaWiki. O MediaWiki tem uma arquitectura baseada nos componentes apresentados no diagrama 4.4.

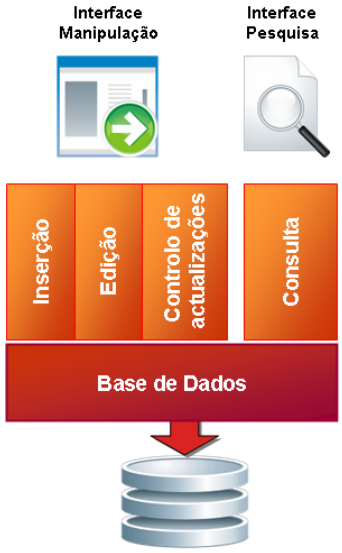


Figura 4.4: *Base de Conhecimento*: Arquitectura e funcionalidades

Através desta interface é possível inserir informação directamente na Base de Conhecimento, bem como editar informação já existente, ter um controlo das actualizações ou simplesmente consultar a informação já existente.

Knowledge Desk Component

Este é o componente principal que faz a ligação entre o *Service Desk* e a Base de Conhecimento e integra estes dois componentes, fazendo a relação entre os incidentes reportados no *Service Desk* para a respectiva solução/dicas na Base de Conhecimento, permitindo ao utilizador pesquisar por artigos relacionados na Base de Conhecimento através da aplicação de *Service Desk*.

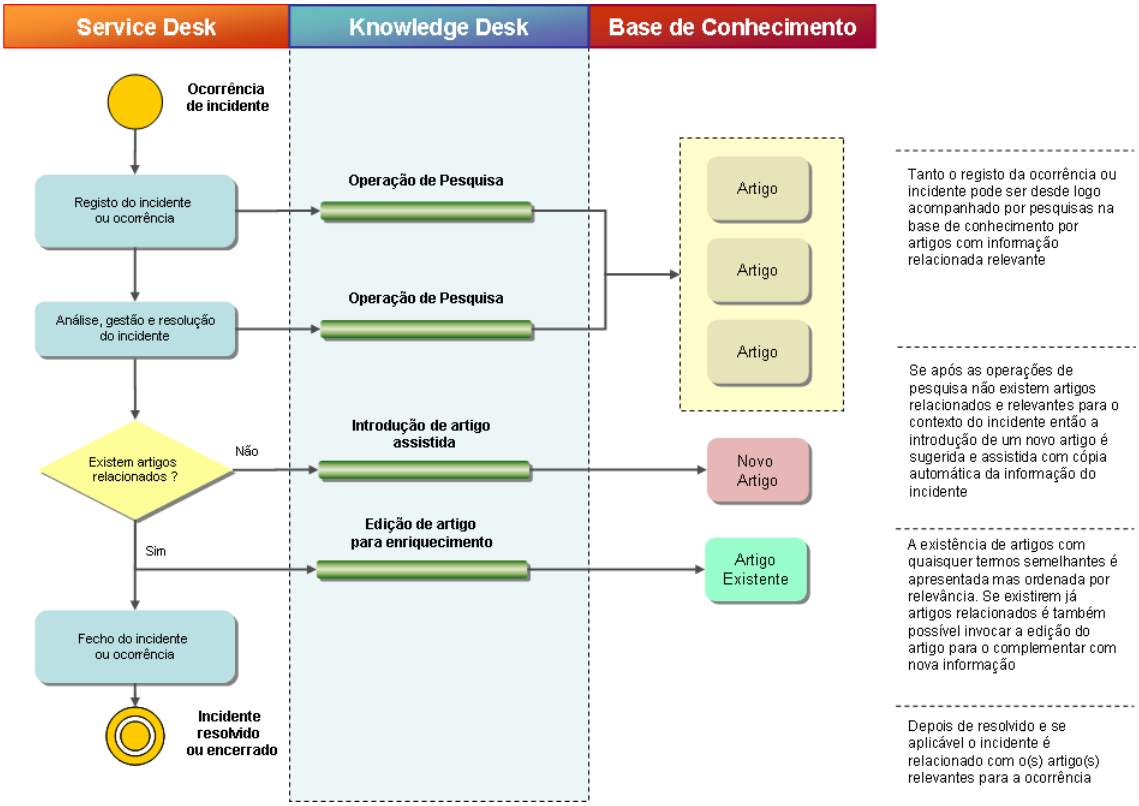


Figura 4.5: *Knowledge Desk Componente*: Diagrama de fluxos de funcionamento

No diagrama 4.5 é apresentado a interação entre os três componentes evidenciando o papel importante do Knowledge Desk. Assim, no registo de incidentes ou ocorrências, pode ser possível a pesquisa na Base de Conhecimento por informação relevante permitindo, desde logo, descortinar se já existe informação relevante para o contexto do incidente ou se será necessário criar um novo artigo.

4.3 Interface Web

Esta camada permite aos utilizadores do sistema consultarem e introduzirem informação na base de dados de uma forma simples, cómoda e eficiente. Por inerência das aplicações selecionadas e também pelos requisitos apresentados a interface do KD é disponibilizada via *browser* o que garante:

- Facilidade de utilização: um utilizador sem conhecimentos da arquitectura e detalhes da implementação pode usar o sistema facilmente;
- Eficiência: a interacção com o utilizador é efectuada de modo a minimizar os tempos de espera e de aprendizagem;
- Acesso: é possível aceder à interface em qualquer local, a partir de qualquer dispositivo, independentemente do tipo de plataforma (hardware/software) em que o utilizador se encontra.

4.3.1 Módulo *Service Desk*

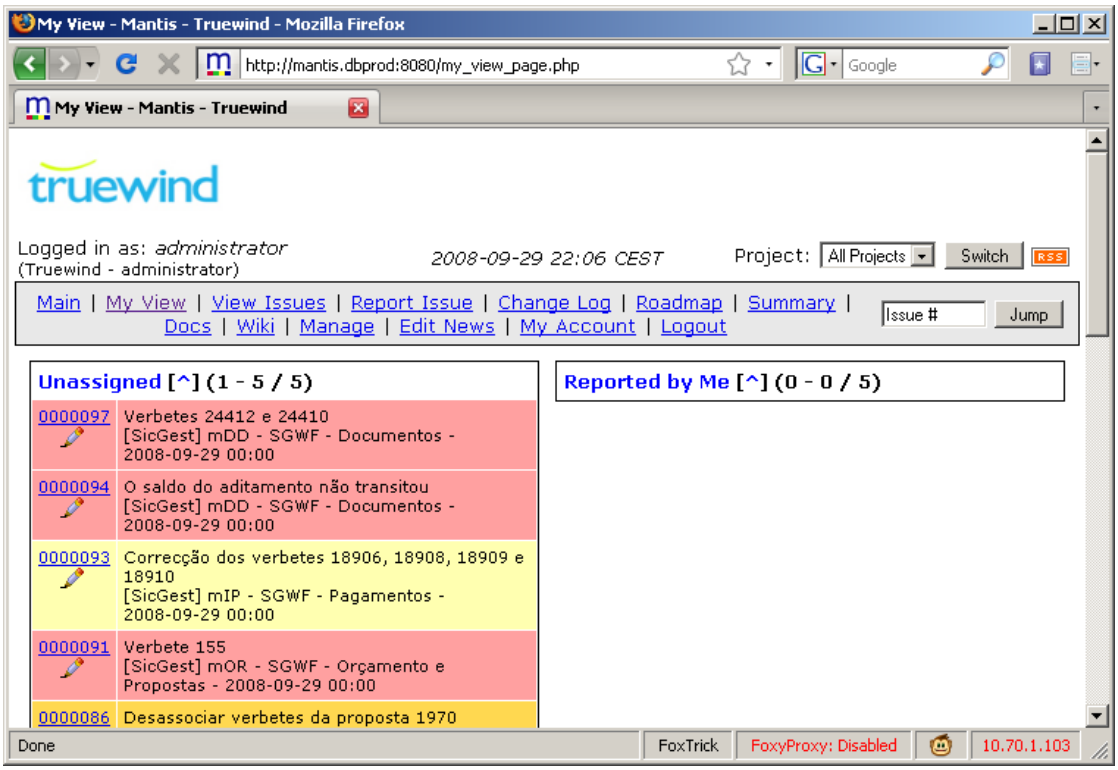


Figura 4.6: Ecrã inicial do Mantis

A interface do *Service Desk* é herdada da aplicação Mantis e é baseada em páginas dinâmicas que produzem HTML através da linguagem PHP. A figura 4.6 ilustra um ecrã desta aplicação.

A interface do *Service Desk* inclui ainda extensões para implementar um conjunto de funcionalidades que incluem a administração e toda a manipulação de incidentes, dando total poder ao utilizador de adicionar informação na Base de Conhecimento ao mesmo tempo que está a registar ou resolver um incidente no *Service Desk*.

4.3.2 Módulo Base de Conhecimento

A Base de Conhecimento, herdada da aplicação MediaWiki, tal como na anterior é baseada em páginas dinâmicas definidas em PHP que, por sua vez, geram HTML. Possui um módulo de pesquisa avançada, bem como os artigos divididos por *namespaces* e categorias facilitando a pesquisa nestes. A figura 4.7 ilustra um ecrã desta aplicação.

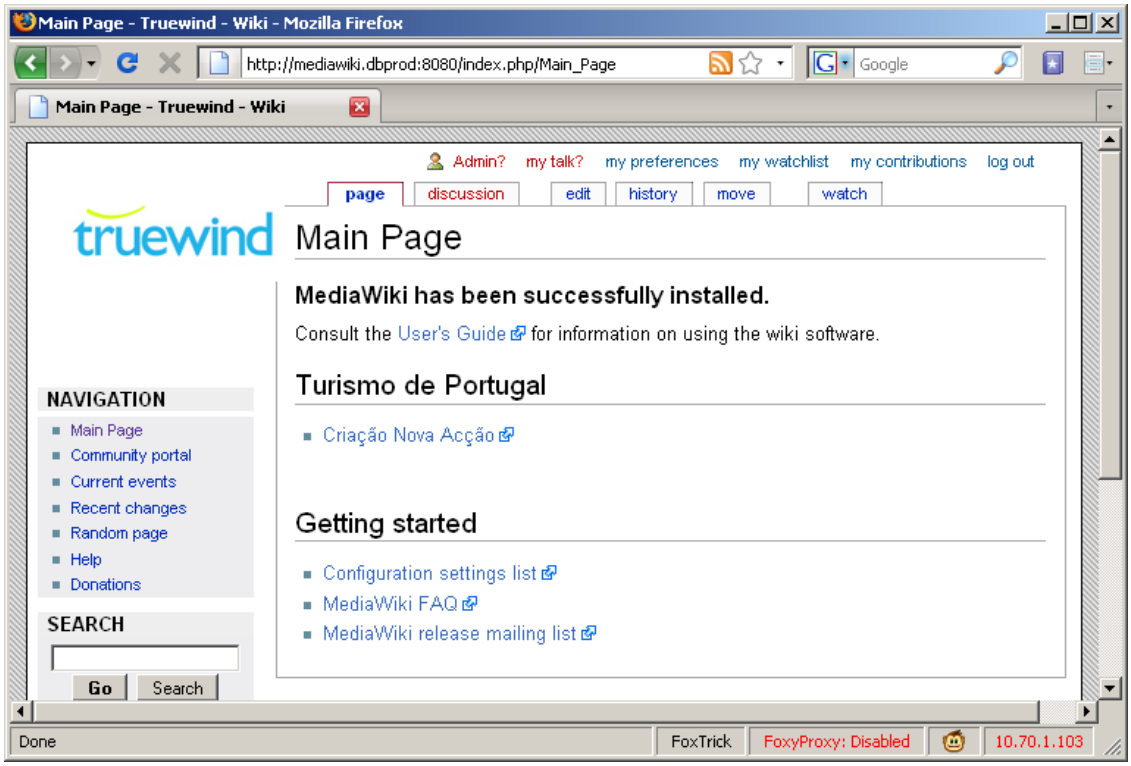
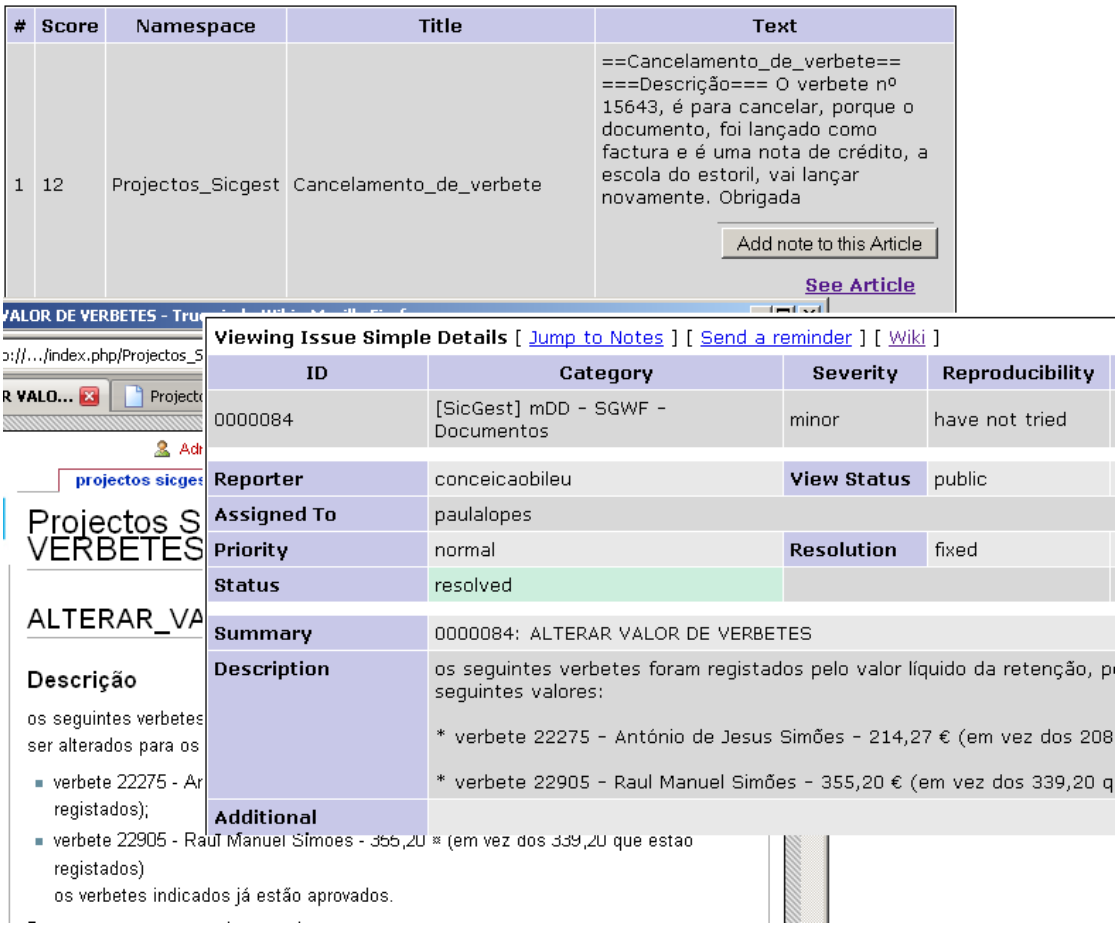


Figura 4.7: Ecrã inicial da MediaWiki

4.3.3 Knowledge Desk

De forma a concretizar a integração entre as duas aplicações, as interfaces das aplicações de *Service Desk* e Base de Conhecimento seleccionadas, passaram a incluir as funções da KD que permitem a pesquisa na MediaWiki e a introdução de artigos no momento do registo e manipulação dos incidentes no Mantis. A figura 4.8 ilustra os vários ecrãs da aplicação desenvolvida.



#	Score	Namespace	Title	Text
1	12	Projectos_Sicgest	Cancelamento_de_verbete	==Cancelamento_de_verbete== ===Descrição=== O verbete nº 15643, é para cancelar, porque o documento, foi lançado como factura e é uma nota de crédito, a escola do estoril, vai lançar novamente. Obrigada

Add note to this Article

See Article

Viewing Issue Simple Details [Jump to Notes] [Send a reminder] [Wiki]

ID	Category	Severity	Reproducibility
0000084	[SicGest] mDD - SGWF - Documentos	minor	have not tried
Reporter	conceicaobileu	**View Status**	public
Assigned To	paulalopes		
Priority	normal	**Resolution**	fixed
Status	resolved		
Summary	0000084: ALTERAR VALOR DE VERBETES		
Description	os seguintes verbetes foram registados pelo valor líquido da retenção, p seguintes valores: * verbete 22275 - António de Jesus Simões - 214,27 € (em vez dos 208 * verbete 22905 - Raul Manuel Simões - 355,20 € (em vez dos 339,20 q		
Additional			

Figura 4.8: Ecrã Knowledge Desk

Capítulo 5

Trabalho Realizado e Resultados

5.1 Trabalho Realizado

O desenvolvimento da KD iniciou-se com a selecção do software a utilizar. Após esta selecção, e antes de iniciar os trabalhos de implementação, foi necessário proceder à instalação, teste e avaliação de software, que incluiu, entre outros: sistema operativo, base de dados, servidor *Web*, ferramentas de desenvolvimento, instalação e teste das aplicações escolhidas.

Só após esta configuração, teste e validação do ambiente criado foi possível iniciar o desenvolvimento dos vários módulos de software que, quando finalizados, implementavam as funcionalidades descritas conceptualmente na arquitectura de sistema.

A plataforma base escolhida para o desenvolvimento e integração dos diversos componentes que constituem Knowledge Desk foi o Linux. Esta escolha foi efectuada por vários motivos:

1. **Baixo Custo**: Não existem custos associados a licenças para a execução do sistema operativo e maior parte dos programas que o acompanham;

2. **Estabilidade**: É um sistema operativo que não necessita de ser reinicializado periodicamente para manter níveis de performance elevados. Além disso, a instalação e actualização de software não necessita, tipicamente, de uma reinicialização do sistema;

3. **Compatibilidade**: Suporta diversos formatos de ficheiros e tecnologias permitindo a interoperabilidade entre elas;

4. **Universalidade**: É o sistema operativo mais utilizado pela comunidade que desenvolve sistemas "open-source" e em que estão disponíveis muitos pacotes para a instalação destas aplicações.

Definida a plataforma base, procedeu-se à sua instalação e configuração.

Para a instalação e configuração do servidor, foi executado o configurador automático que é incluído no pacote instalado e que, com algumas definições simples, configura o servidor para disponibilizar toda a informação que seja passível de ser obtida.

Foi instalada a base de dados Oracle sob a plataforma base, parte essencial do componente de armazenamento de dados da arquitectura definida.

Após a instalação dos pacotes de software base mais relevantes, foram instaladas versões customizadas do Apache (*Web server*) e das bibliotecas PHP, necessárias para a camada de interface com o utilizador. A obrigação de criar uma versão customizada destes componentes da infra-estrutura deveu-se à indisponibilidade de uma distribuição de PHP/Apache compilada com as bibliotecas de suporte para ligação a uma base de dados Oracle. Para isto, foi necessário obter o código fonte dessas aplicações e proceder à compilação, instalação e configuração das mesmas. Como consequência desta actividade, a aplicação de actualizações pré-compiladas em distribuição aos pacotes base está limitada.

5.1.1 Tecnologias Utilizadas

A identificação das tecnologias mais adequadas para o desenvolvimento de um sistema como o KD, é um factor chave no sucesso do desenvolvimento do sistema. Depois de identificadas as tecnologias candidatas, que incluíram universos tão distantes como Microsoft ou Java, foi adoptado um conjunto de tecnologias universalmente aceite para o desenvolvimento de soluções deste tipo. Das fases de familiarização e adaptação às tecnologias seleccionadas resultou um conjunto de informação relevante que se inclui na tabela 5.1 (na página 42) e que expõe as vantagens e desvantagens de cada tecnologia envolvida na construção da KD e na forma como foi sentida a sua utilização no processo de desenvolvimento da solução.

Com todo o software instalado e configurado e com o domínio da tecnologia base para o desenvolvimento da solução, deu-se início ao desenvolvimento dos módulos necessários para cada um dos componentes da arquitectura.

5.1.2 Aplicações de Service Desk

A primeira iteração executada no desenvolvimento do projecto envolveu o desenvolvimento deste componente que teve início com a instalação da aplicação Mantis. A aplicação foi instalada com base de dados MySQL de maneira a ser possível obter os passos da instalação e qual o resultado final, ao nível de ficheiros, tabelas, *triggers* [1] e índices criados. A instalação deste componente foi feita via *browser*, sendo

[1] Trigger ou Gatilho é um procedimento executado de forma automática cada vez que ocorrem determinados eventos numa tabela em particular, de uma base de dados.

criado automaticamente o esquema da base de dados e o ficheiro *"config_inc.php"* onde ficam guardados os dados mais sensíveis e essenciais ao bom funcionamento do software tais como o utilizador e palavra-chave de acesso à base de dados. Finda a instalação, passou-se ao objectivo seguinte, a criação e integração do Mantis com base de dados Oracle.

Ao instalar o Mantis é possível escolher base de dados Oracle, como suporte, ao invés de MySQL, apontado por omissão. No entanto, é emitida uma advertência pois a opção Oracle é experimental. É também possível ao utilizador escolher apenas para imprimir no ecrã o *script* de criação da base de dados.

Efectuando estes passos manualmente foi possível verificar algumas incorrecções, ao nível da criação de *triggers* e do tamanho de alguns campos, nomeadamente no nome de algumas tabelas que excediam o número máximo de 30 caracteres permitido pelo software da Oracle.

Prosseguindo com os trabalhos de configuração do mantis para uso do Oracle como base de dados, procedeu-se à alteração do ficheiro *"config_inc.php"*, onde se modificou o tipo de base de dados de *mysql* para *ora8*, o nome da bd, o utilizador e a palavra-chave. Foi também necessário modificar, no código, o nome das tabelas alteradas em cada local em que estas eram referenciadas, dado a alteração ao nome destas, para uma dimensão inferior a 30 caracteres, o que acabou por ser mais fácil do que se poderia imaginar ao início, pois o ficheiro de configuração contém variáveis globais associadas ao nome das tabelas criadas na base de dados. Foi também modificado o procedimento para evitar o uso de caracteres especiais, no acesso à base de dados Oracle, como a plica, que pode ser utilizada para aceder de forma não autorizada ao sistema. No código usado, é verificado primeiro se a variável de configuração do PHP, `magic_quotes_sybase` se encontra a verdadeiro pois, caso isto aconteça, a função PHP `addslashes`, em vez de adicionar uma contra-barra, adiciona uma plica antes de cada caracter especial, caracteres estes que são as plicas, aspas, contra-barras e o byte `NULL`. A função criada, bem como aspectos essenciais da configuração são apresentados de seguida:

```
# --------------------
# prepare a string before DB insertion
# @@@ should default be return addslashes( $p_string );
# or generate an error
# @@@ Consider using ADODB escaping for all databases.
function db_prepare_string( $p_string ) {

 (...)
```

```
 case 'oci8':

    //$t_escaped = $g_db->qstr( $p_string, true );
    //return $t_escaped;
    //break;

    # Returns the value of the configuration option as a string
    # on success, or an empty string on failure or for null values.
    # magic_quotes_sybase - If enabled, a single-quote is escaped
    # with a single-quote instead of a backslash.
    # If on, it completely overrides
    if( ini_get( 'magic_quotes_sybase' ) ) {

            return addslashes( $p_string );
    } else {

            ini_set( 'magic_quotes_sybase', true );
            $t_string = addslashes( $p_string );
            ini_set( 'magic_quotes_sybase', false );
            return $t_string;
    }

(...)

}
```

Apesar do tempo despendido nesta iteração, o objectivo foi cumprido e a passagem do Mantis para base de dados Oracle foi relativamente fácil, dado o facto deste usar uma abstracção ao nível da comunicação com a base de dados. Esta é suportada pela *framework* [2] AdoDB [1], desenvolvida em PHP, com suporte a uma já grande variedade de SGBD e já exaustivamente testada.

5.1.3 Aplicação de Base de Conhecimento

A segunda iteração do trabalho consistiu em avaliar, instalar e integrar com base de dados Oracle a componente da Base de Conhecimento. Para esta etapa houve duas aplicações que suscitaram interesse e se achou serem mais indicadas indo de encontro aos requisitos da arquitectura, a DokuWiki e a MediaWiki.

[2]Framework é um conjunto de bibliotecas, *scripts* ou classes especializadas numa determinada função/responsabilidade para auxiliar e unir diferentes componentes de um projecto de software.

A escolha acabou por recair, inicialmente, sobre a DokuWiki fundamentalmente por ser mais "leve"que a MediaWiki, mesmo apesar desta ser baseada em ficheiros no armazenamento da informação, não baseando, por isso, o seu funcionamento num SGBD relacional. A ideia passava por criar um esquema de tabelas adequado à DokuWiki e usar a *framework* usada pelo Mantis, *AdoDB*, para criar a abstracção de ligação à base de dados, permitindo aproximar as duas aplicações, a DokuWiki e o Mantis. No entanto, esta tarefa acabou por ser mais complexa e morosa do que se pretendia inicialmente, acabando, a DokuWiki, por não ser a aplicação ideal.

Passou-se então à instalação do software usado pela Wikipédia, a MediaWiki, escolhida pelas suas provas dadas, entre elas, a robustez. Tal como na instalação do Mantis, iniciou-se a instalação da aplicação com suporte MySQL, pois é o SGBD usado por omissão por esta aplicação, para uma melhor perspectiva do funcionamento e familiarização com todo o processo de instalação, os passos a seguir e os ficheiros gerados. Finalizado este processo de ambientação seguiu-se a migração do SGBD de MySQL para Oracle.

Apesar da MediaWiki já suportar Oracle não é exequível escolher esta opção aquando da instalação. No entanto, na directoria `maintenance`, é possível encontrar alguns ficheiros já com as devidas alterações para a criação do esquema de base de dados Oracle. Após a execução do *script*, foi necessário alterar os dados de acesso à base de dados no ficheiro de configuração, neste caso, *"LocalSettings.php"*.

Após estas alterações a aplicação passou a reportar erros de forma sistemática sempre que se verificava uma consulta à base de dados. O problema veio-se a verificar no *script* de acesso à base de dados criado para o SGBD Oracle, *"DatabaseOracle.php"*, onde foram alterados uns quantos pormenores significativos, sendo que o mais importante foi a alteração da seguinte função:

```
function fetchRow($res) {
    # NS :: 20080403
    # Acrescentado o if a funcao original
    if ($res instanceof ResultWrapper) {
        #$res = $res->fetchAssoc();
        $res = $res->result;
    }

    $res = $res->fetchAssoc();
    return $res;
}
```

Apesar dsta alteração parecer de todo trivial, foi aqui aplicado um período de tempo superior ao que estava previsto, para se conseguir determinar a anomalia, muito devido ao total desconhecimento da aplicação, das suas classes e código. Para além disso, ao mesmo tempo em que se percorre o código para localizar os problemas e anomalias, foram verificados os mecanismos de segurança ao nível dos tipos de utilizadores e as suas permissões para implementação das funcionalidades e controlo de actualizações própria da aplicação de Base de Conhecimento. Como tal, foi preciso ter em atenção os mecanismos de autenticação já desenvolvidos pela aplicação MediaWiki e a forma de bloquear a aplicação a utilização não autorizada. Isto foi conseguido adicionando algumas linhas ao ficheiro de configuração:

```
// Implicit group for all visitors
// $wgGroupPermissions['group']['right'] = true /* or false */;
$wgGroupPermissions['*'    ]['createaccount']    = true;
$wgGroupPermissions['*'    ]['read']             = false;
$wgGroupPermissions['*'    ]['edit']             = false;
$wgGroupPermissions['*'    ]['createpage']       = false;
$wgGroupPermissions['*'    ]['createtalk']       = false;
```

Outra questão que foi tida em conta, foi a estrutura que seria adoptada na organização da informação, quais os *namespaces* a criar e como os criar. Na MediaWiki, os *namespaces* são criados adicionando-os à variável global `$wgExtraNamespaces` no ficheiro de configuração. Para evitar perder esta informação e se poder editar sem restricções os *namespaces* criados decidiu-se, ao invés de os criar no ficheiro de configuração, criar um *script* novo que foi depois incluído no ficheiro de configuração através a linha `require_once( "$IP/prx_namespaces.php" );`. Assim, o *script* `prx_namespaces.php` ficou com o seguinte aspecto:

```
<?php
# Mantis - a php based bugtracking system

# Copyright (C) 2008 Práxia SI   - nuno.salvador@praxia.pt

# Mantis is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
```

```
# Mantis is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with Mantis.  If not, see <http://www.gnu.org/licenses/>.

# --------------------------------------------------------------------
# $Id: wiki_component.php,v 1.0.0.5 2008-05-05 17:18:00 nsalvador Exp $
# --------------------------------------------------------------------

# Don't change comments on this file
# wiki_component.php depends on it.
?>
<?php
# CONSTANTS
define("MANTIS", 100);
define("MANTIS_TALK", 101);
define("PROJECTOS", 102);
define("PROJECTOS_TALK", 103);
define("TECNOLOGIA", 104);
define("TECNOLOGIA_TALK", 105);
define("SCRIPTS", 106);
define("SCRIPTS_TALK", 107);
define("PROJECTOS_SUCH", 108);
define("PROJECTOS_SUCH_TALK", 109);
define("PROJECTOS_OGMA", 110);
define("PROJECTOS_OGMA_TALK", 111);
#VAR
$wgExtraNamespaces[MANTIS] = "Mantis";
$wgExtraNamespaces[MANTIS_TALK] = "Mantis_Talk";
$wgExtraNamespaces[PROJECTOS] = "Projectos";
$wgExtraNamespaces[PROJECTOS_TALK] = "Projectos_Talk";
$wgExtraNamespaces[TECNOLOGIA] = "Tecnologia";
$wgExtraNamespaces[TECNOLOGIA_TALK] = "Tecnologia_Talk";
$wgExtraNamespaces[SCRIPTS] = "Scripts";
$wgExtraNamespaces[SCRIPTS_TALK] = "Scripts_Talk";
$wgExtraNamespaces[PROJECTOS_SUCH] = "Projectos_Such";
```

```
$wgExtraNamespaces[PROJECTOS_SUCH_TALK] = "Projectos_Such_Talk";
$wgExtraNamespaces[PROJECTOS_OGMA] = "Projectos_Ogma";
$wgExtraNamespaces[PROJECTOS_OGMA_TALK] = "Projectos_Ogma_Talk";
#CONTENT_NS
$wgContentNamespaces[] = MANTIS;
$wgContentNamespaces[] = PROJECTOS;
$wgContentNamespaces[] = TECNOLOGIA;
$wgContentNamespaces[] = SCRIPTS;
$wgContentNamespaces[] = PROJECTOS_SUCH;
$wgContentNamespaces[] = PROJECTOS_OGMA;
#FIM
```

5.1.4 Knowledge Desk Component

Por fim, o desenvolvimento da Knowledge Desk, em si, isto é, a plena integração de uma aplicação de *Service Desk* com uma Base de Conhecimento, neste caso do Mantis com a MediaWiki, foi realizado de acordo com ciclos de objectivos curtos e concretos.

O primeiro grande objectivo, passou por conseguir adicionar de forma automática toda a informação introduzida no Mantis na MediaWiki, deixando para segundo plano as preocupações a nível de contexto e redundância. Assim, cada novo incidente criado no *Service Desk*, passou a ser um novo artigo na Base de Conhecimento e, sempre que uma nota era adicionada ao incidente no *Service Desk*, esta também era adicionada ao respectivo artigo. Nesta fase foi também criada a tabela `Mantis_Wiki_Table` que permitiu guardar as associações entre os incidentes e artigos criados.

Depois de cumprido o objectivo de ser possível adicionar informação na Base de Conhecimento através do *Service Desk*, passou-se ao segundo objectivo de tentar não adicionar informação redundante na Base de Conhecimento e de adicionar notas referentes a um incidente do *Service Desk* a outro artigo diferente na Base de Conhecimento. Para alcançar este objectivo, foi estudada a possibilidade de usar o *Oracle Text*. O *Oracle Text* permite a criação de estruturas de indexação de texto na base de dados e a utilização de funções de recuperação de informação com base nessas estruturas. Aqui, o grande desafio deparava-se em conseguir introduzir da melhor maneira as funcionalidades oferecidas por esta ferramenta. A maneira encontrada, com o suporte do *Oracle Text*, foi a apresentada em baixo, como parte de uma pesquisa SQL,

```
$query = "SELECT SCORE(1), pc.old_text, page.page_title,
            page.page_namespace, page.page_id
```

```
        FROM ( ( pagecontent pc LEFT JOIN
                    revision rev on rev.rev_text_id = pc.old_id )
         LEFT JOIN page on rev.rev_page = page.page_id
        )
  WHERE rev.rev_id = page.page_latest
  AND CONTAINS(old_text, 'about(". $t_bugnote .")', 1) > 0
ORDER BY SCORE(1) DESC";
```

onde o `CONTAINS(old_text, 'about(". $t_bugnote .")', 1) > 0`, permite-nos pesquisar todos os artigos na coluna `old_text` da tabela `pagecontent` que contenham as palavras dadas na variável `$t_bugnote`. Os artigos são depois apresentados ao utilizador ordenados pelos mais favoráveis, isto é, aqueles com maior probabilidade de serem relevantes ao que procuramos para aqueles onde as ocorrências das palavras dadas são menores.

Depois desta fase passou a ser possível introduzir, de maneira selectiva, a informação na MediaWiki, através do Mantis. Isto é, o utilizador continuava a criar uma entrada na Base de Conhecimento por cada incidente que registava mas, ao adicionar uma nota num incidente já registado podia fazer uma pesquisa por todos os artigos da Base de Conhecimento à procura das entradas semelhantes à nota que fosse introduzir, possibilitando a adição a informação a um outro artigo na Base de Conhecimento que não o artigo pertencente àquele incidente específico.

O terceiro objectivo, passou por dar mais liberdade ao utilizador não criando automaticamente um artigo na Base de Conhecimento por cada incidente criado no Mantis, mas permitindo ao utilizador escolher se pretende, ou não, que o incidente a criar seja introduzido na Base de Conhecimento, ao mesmo tempo que permite pesquisar por artigos semelhantes e dando-se ainda a hipótese de adicionar a informação do novo incidente a um artigo já existente.

Por fim, é dado um total controle sobre a inserção dos artigos na Base de Conhecimento ao utilizador, podendo este criar artigos novos e/ou adicionar artigos/notas a artigos já existentes, foram também melhorados pormenores de interface e aplicação ao nível da personalização. Um problema residia no facto dos artigos na Base de Conhecimento serem criados com o nome do incidente registado no *Service Desk*, não sendo claramente este o melhor método. Neste caso o utilizador podia criar, por exemplo, dois incidentes redundantes e duplicados, *"user_bloqueado23"* e *"user_bloqueado"*. Caso o utilizador escolhesse criar estes artigos, ou apenas um deles na Base de Conhecimento, o primeiro não era, de todo, o nome mais adequado a dar. Uma opção mais inteligente seria o de dar, por exemplo, o nome *"user_bloqueado"*

apenas criando assim um artigo genérico com a solução para todos os bugs abertos para este mesmo problema. Assim, foi concretizado um módulo do KD que, através do uso de tecnologia AJAX, apresenta ao utilizador uma sugestão para o nome do artigo podendo, no entanto, este editar o nome para um mais adequado. O código que concretiza esta funcionalidade é exposto de seguida:

```
<script type="text/javascript" language="JavaScript" >
<!--
function dual_submit(isContext){
   if ( isContext == true ) {
          document.report_bug_form.action
= "bug_context_page.php?type=<?php echo REPORT ?>";
          document.report_bug_form.target = "_blank";
          document.report_bug_form.submit();
   } else {
          document.report_bug_form.action = "bug_report.php";
          document.report_bug_form.target = "_self";
          document.report_bug_form.submit();
   }
}//dual_submit();

function escolheNome() {
   if ( !document.report_bug_form.wiki_create.checked ) {
     document.getElementById('article_name').style.display = "none";
   }//if
   else {
     //var summary = document.getElementsByName('summary');
     var s = document.report_bug_form.summary.value;
     var p_id = <?php echo $t_project_id; ?>;
     new Ajax.Updater('xml', 'wikie_article_name_request.php',
         {
            method: 'get',parameters:
            {
             project_id: p_id, summary: s},
             onComplete: function f(){ workName();
            }
         }
      );
```

```
   }//else

}//escolheNome();

function workName() {
  var xmlElement = $('result');
  var clean_result = xmlElement.title;
  var replaceElement = $('replace');
  replaceElement.value = clean_result;
  document.getElementById('article_name').style.display = "";
}//workName();
-->
</script>

<?php if( pageExists( $f_bug_id ) !== false ) { ?>
<!-- ACTUALIZAR WIKI COM ESTA NOTA -->
<tr <?php echo helper_alternate_class() ?>>
    <td class="category">
        <?php echo lang_get( 'wiki_add_note' ) ?> ?
    </td>
    <td>
        <input type="checkbox" name="add_note"
                        <?php check_checked( $f_add_note ) ?>
        /> (<?php echo lang_get( 'check_add_wiki' ) ?>)
    </td>
</tr>

<?php } else { ?>

<!-- Wiki Create Article -->
<tr <?php echo helper_alternate_class() ?>>
    <td class="category">
        <?php echo lang_get( 'wiki_create_article' ) ?> ?
    </td>
    <td>
        <input type="checkbox" name="wiki_create"
                    <?php check_checked( $f_wiki_create ) ?>
                    onClick="escolheNome();" />
                    (<?php echo lang_get( 'check_create_wiki' )
```

```
        ?>)
    </td>
</tr>

<!-- Escolher Nome para o Artigo a criar -->
<tr style="display:none" id="article_name"
    <?php echo helper_alternate_class() ?>
>
    <td class="category">
        <?php echo lang_get( 'wiki_article_name' ) ?>
    </td>
    <td>
        <div id="xml">
        </div>
        <input id="replace" type="text" name="article_name_text"
                    size="60" maxlength="128" value="" />
                    (<?php echo lang_get( 'check_article_name' )
        ?>)
    </td>
</tr>
<?php }//else ?>


<?php

   require_once( 'core.php' );
   $t_core_path = config_get( 'core_path' );

   auth_ensure_user_authenticated();

   require_once( $t_core_path . 'wiki_component.php' );

   $c_project_id = array();
   $c_project_id = gpc_get_int( 'project_id' );
   $c_summary = array();
   $c_summary  = gpc_get_string( 'summary' );


   $result = setArticleName( $c_project_id, $c_summary );
```

```
   header( 'Content-type: application/xml; charset="utf-8"', true );
   echo '<?xml version="1.0" encoding="ISO-8859-1"?>';
   echo '<label id="result" title="'. $result . '"></label>';

?>

<?php
/**
 * Funcao que dado aconselha ao utilizador
 * um namespace e nome para o Artigo a criar
 *
 * @param p_project_id : o id do projecto onde
 *                       o bug esta a ser criado no mantis
 * @param p_summary : o sumário do bug a ser criado no mantis
 *
 * @return o namespace:nome_do_artigo
 */
function setArticleName($p_project_id, $p_summary) {

   $c_summary     = $p_summary;
   $c_project_id  = $p_project_id;

   $c_summary     = ucfirst( $c_summary );
   $c_desc_temp   = '=='.$c_summary.'=='. "\r\n";
   $c_summary     = str_replace( ' ', '_', $c_summary );

   $c_page_namespace
        = getNamespaceFromId( intval( getNamespace( $c_project_id )));

   return $c_page_namespace.":".$c_summary;
}

?>
```

Outro problema encontrado residia nos facto dos *namespaces* terem que ser criados manualmente no ficheiro `anexos/prx_namespaces.php`. Isto implicava que o

utilizador responsável pela aplicação, editasse manualmente o ficheiro, com a sintaxe correcta, para criar um projecto na Base de Conhecimento cada vez que criava um projecto no *Service Desk*. Para evitar algum tipo de esquecimento e entropia na base de dados, cada vez que um incidente é adicionado na Base de Conhecimento e se o projecto ainda não existe na Base de Conhecimento, este é criado de maneira correcta, pela KD, no ficheiro.

Finalmente um o utro problema surgido no desenvolvimento da KD residia no facto de criar ligações directas para os artigos criados na Base de Conhecimento, quando estes eram criados na criação/actualização de um incidente. A aplicação Mantis possui um *link* na página de cada incidente criado para ligar à Base de Conhecimento. Mas, este, limita-se a abrir o endereço da Base de Conhecimento com o identificador do incidente no Mantis. Dado que o utilizador tem liberdade de escolher o nome com que o artigo é criado na Base de Conhecimento, incluindo o *namespace*, foi necessário alterar esta funcionalidade no código. Dado que a informação sobre a correspondência entre os incidentes e os artigos se encontra na tabela `Mantis_Wiki_Table` apenas foi necessário alterar o código dinâmico que gera o HTML para o *link* apontar para o artigo correcto.

5.2 Resultados Alcançados

O protótipo, resultado do processo de desenvolvimento, foi instalado e utilizado no apoio ao projecto a ser desenvolvido pela Práxia no "Turismo de Portugal - IP"e permitiu aumentar a eficácia e produção neste cliente sendo que até ao momento foram registados na aplicação cerca de uma centena de incidentes e 7 artigos, dado que muitos dos incidentes reportados são reincidentes.

De acordo com o relato do utilizador responsável pelo suporte ao cliente, a KD supera a anterior solução baseada unicamente no software Mantis nos seguintes aspectos:

- permite a pesquisa de soluções por problemas já reportados;
- permite a resolução por parte de um utilizador não habituado a resolver um dado problema, através da utilização da Base de Conhecimento.

Segundo o mesmo utilizador, a KD ainda apresenta margem para melhorias nos seguintes aspectos:

- os artigos criados através do *Service Desk* deveriam ter um *link* para o respectivo incidente;
- deveria ser possível obter uma listagem dos artigos criados na Base de Conhecimento;

- no *Service Desk* quando se altera o estado do incidente para "resolvido", são registadas as acções de resolução no *Service Desk* mas estas não são acrescentada ao artigo na Base de Conhecimento. Deverão ser replicadas estas alterações para a Base de Conhecimento.

A figura 5.1 ilustra a criação de um incidente no Mantis com indicação de criação de artigo na Base de Conhecimento na instalação em funcionamento no "Turismo de Portugal".

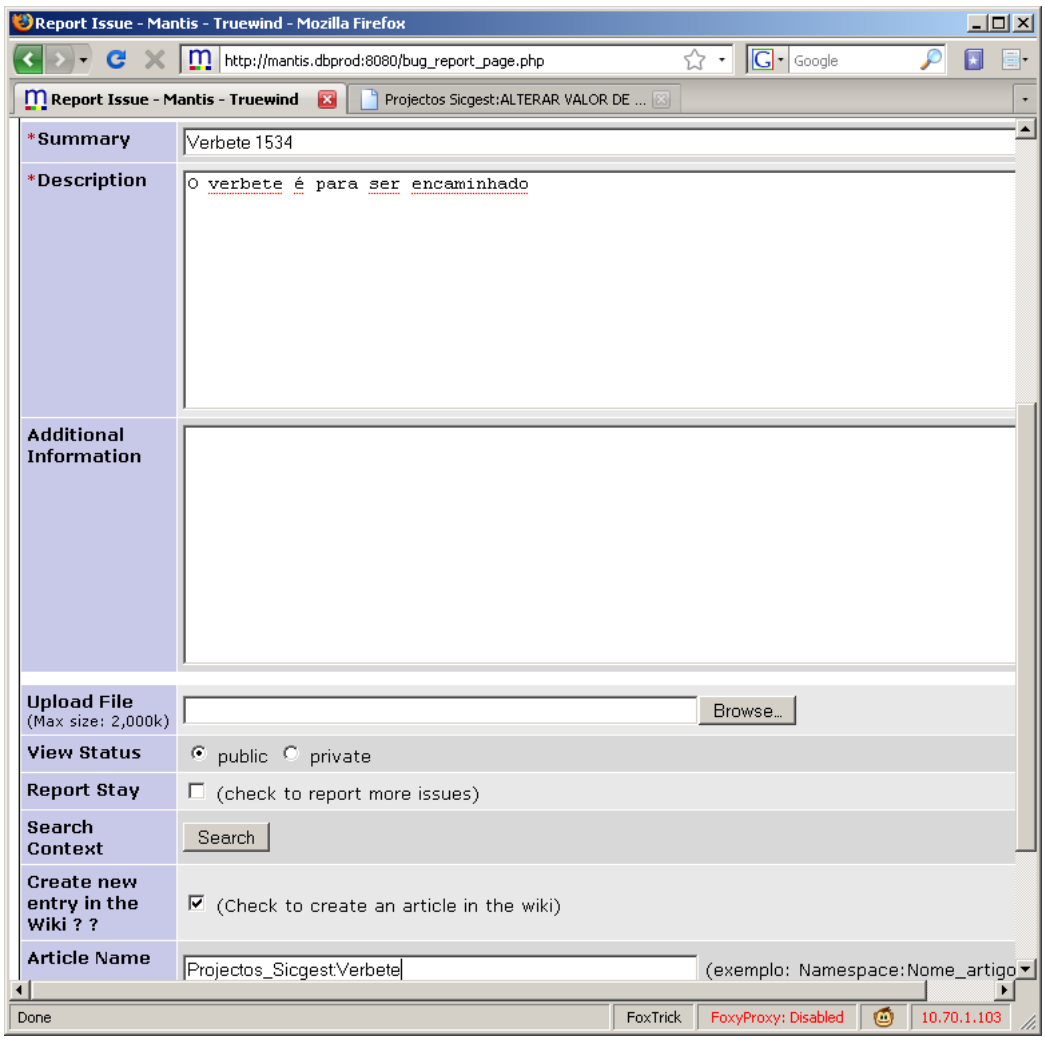


Figura 5.1: Criação de um incidente, adicionando à Base de Conhecimento

Tecnologia	Pontos Fortes	Pontos Fracos	Aplicado em:
PHP	- Open-Source - Multiplataforma - Fácil de Programar - Interfaces do Mantis e MediaWiki - Boa documentação	- Pouco Modular - Linguagem interpretada em tempo de execução	- Interface Web
Oracle Text	- Rápida indexação dos artigos - Integrado na Base de Dados - Fácil de utilizar e programar em SQL	- Ocupa muito espaço em disco - Requer acções de administração para que a actualização dos índices seja automática	- Pesquisa na base de dados
AJAX	- Requer menos largura de banda - Interfaces mais interactivas - Evita ter que carregar a página toda	- Programação Complexa - Interpretado de maneira diferente nos vários *browsers* - Acessibilidade	- Interface Web
XML	- Permite definir uma linguagem mais precisa para um problema - Separação entre conteúdo e apresentação	- Mais complexo e exigente que HTML - Requer processamento para apresentação dos conteúdos	- Knowledge Desk Component
MySQL	- Custo - Código aberto - Portabilidade - Desempenho, Robustez, Multi-tarefa e Multi-usuário	- Funções de manipulação de dados avançadas	- Instalação por omissão da Mediawiki - Instalação por omissão do Mantis
Oracle	- Information retrieval - Usada pelos clientes da Práxia	- Custo	- Base de dados da KD

Tabela 5.1: Pontos Fortes e Fracos das Tecnologias Envolvidas

Capítulo 6

Conclusão

O desenvolvimento da KD pretendia criar uma nova aplicação integrando outras duas já existentes, um *Service Desk* e uma Base de Conhecimento, de maneira a facilitar a alimentação da Base de Conhecimento através da resolução de pedidos no *Service Desk*. Consideramos este objectivo alcançado.

A principal dificuldade no seu desenvolvimento foi encontrar uma forma de popular a base de conhecimento, com a informação adquirida com o decorrer da resolução de pedidos no *Service Desk*, de maneira eficiente.

O sucesso do projecto traduziu-se na instalação e utilização da KD no apoio ao projecto a ser desenvolvido pela Práxia no "Turismo de Portugal". Dadas as características e dimensão do TP, em termos aplicacionais, é essencial ter uma ferramenta de controlo onde os utilizadores possam reportar os incidentes encontrados e onde os técnicos possam ir alimentado uma Base de Conhecimento dos problemas já resolvidos para conseguir cada vez mais automatizar todo o processo ganhando, mais tarde, tempo valioso na resolução de problemas semelhantes.

A nível pessoal, o estágio foi uma experiência enriquecedora. A Práxia é uma empresa pequena mas que possui valores muito fortes e objectivos claros orientados à satisfação dos clientes que tem em carteira. O ambiente de trabalho é muito acolhedor e devido à sua dimensão, sente-se que fazemos parte de uma família. Isto reflecte-se nas fortes relações inter-pessoais e espírito de entreajuda que existe entre os colaboradores. Apesar de ser a minha primeira experiência no mercado de trabalho, é com enorme satisfação e orgulho que faço parte da Práxia.

O trabalho nas OGMA foi importante, na medida em que permitiu uma aprendizagem aprofundada da tecnologia Oracle e uma melhor compreensão dos processos e metodologias que regem o mercado de trabalho actual. Por outro lado, o desenvolvimento do projecto permitiu a aprendizagem de novas linguagens e técnicas de programação bem como o aprofundamento de outras com as quais já tinha contacto. Cadeiras como *Sistemas Operativos*, *Sistemas de Informação e Base de Dados*, *Interface Pessoa Máquina*, *Configuração e Gestão de Sistemas*, *Projecto em Sistemas*

de Informação, *Desenvolvimento Centrado em Objectos*, *Segurança* e *Linguagens Formais e Autómatos* foram importantes em todo o desenvolvimento da Knowledge Desk.

O estágio foi uma etapa fundamental e necessária para a minha formação académica e pessoal. Desempenhou um papel importante na minha integração do mercado de trabalho e na aprendizagem de novas tecnologias na área da informática.

6.1 Trabalho Futuro

Apesar do essencial ter sido implementando e dos objectivos propostos terem sido cumpridos, alguns pormenores que tornariam a KD mais funcional e apelativa ficaram para trás, devido aos prazos limitados.

A MediaWiki usa uma sintaxe própria que depois é convertida dinamicamente para HTML. Aquando das pesquisas no Mantis dos artigos na Base de Conhecimento, esta tradução não é feita, sendo apresentando ao utilizador alguns caracteres não legíveis nos artigos, não sendo agradável nem intuitivo. Será portanto necessário, num futuro próximo, criar uma estrutura para realizar estas transformações.

Para além de criar incidentes e acrescentar notas, existem outras acções no *Service Desk* como *"feedback"*, *"confirmed"* que não foram tidas em conta para a população da Base de Conhecimento.

É necessário também rever o código Javascript na alteração dinâmica das `actions` dos `forms` pois este não funciona de forma coerente em todos os browsers.

Outra alteração, a ter em conta no futuro, é o facto do *Service Desk* quando aconselha e tenta adicionar um novo artigo na Base de Conhecimento não fazer a verificação se o título do artigo a adicionar já existe para aquele *namespace*, não advertindo, assim, o utilizador para tal ocorrência, ocorrendo o erro de "artigo já existente" mais tarde.

Por fim, uma última alteração a realizar passa por juntar a criação e *login* dos utilizadores do Mantis e MediaWiki num só, facilitando assim a introdução de credenciais do utilizador.

Acrónimos

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
BI	Business Intelligence
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DBMS	Database Management System
FCUL	Faculdade de Ciências da Universidade de Lisboa
GPL	GNU General Public License
GUI	Graphic User Interface
HTML	HyperText Markup Language
JVM	Java Virtual Machine
KD	Knowledge Desk
OGMA	Oficinas Gerais de Material Aeronáutico
PEI	Projecto em Engenharia Informática
PHP	PHP Hypertext Preprocessor
RDBMS	Relational Database Management System
SD	Service Desk
SGBD	Sistema de Gestão de Base de Dados
SOA	Service-Oriented Architecture
SQL	Structured Query Language
TI	Tecnologias de Informação
TP	Turismo de Portugal

UCM	Oracle Universal Content Management
XML	eXtensible Markup Language
XSS	Cross-Site Scripting

Bibliografia

[1] AdoDB. http://adodb.sourceforge.net/, 2001.

[2] Adventnet. http://manageengine.adventnet.com/products/service-desk, 2008.

[3] Eric J. Braude. *Software Enginnering, An object-Oriented Perspective*. John Wiley and Sons, Inc, 2001.

[4] Bugtracker. http://en.wikipedia.org/wiki/bugtracker, 2008.

[5] CA. http://ca.com/worldwide/, 2008.

[6] Página da HP. http://www.hp.com/, 2008.

[7] Página da Práxia. http://www.praxia.pt, 2008.

[8] CA Service Desk. http://ca.com/us/service-desk.aspx, 2008.

[9] Ilient Service Desk. http://www.ilient.com/, 2008.

[10] Página do PHP. http://php.net, 2008.

[11] DokuWiki. http://wiki.splitbrain.org/wiki:dokuwiki, 2008.

[12] David Flanaganv. *JavaScript: The Definitive Guide*. O'Reilly, 2001.

[13] Free Software Foundation. Gpl - http://www.gnu.org/licenses/gpl.html, 2008.

[14] Free Software Foundation GNU. http://www.gnu.org/, 2008.

[15] Alex Kriegel and Boris M. Trukhnov. *SQL Bible*. John Wiley and Sons, 2003.

[16] Hakon Wium Lie and Bert Bos. Cascading style sheet - designing for the web, 2005.

[17] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, http://www-csli.stanford.edu/ hinrich/information-retrieval-book.html, 2008.

[18] Mantis. http://www.mantisbt.org/, 2008.

[19] MediaWiki. http://www.mediawiki.org/, 2008.

[20] Meta4. http://www.meta4.com/, 2008.

[21] MySQL. http://www.mysql.com/, 2008.

[22] Oracle. http://www.oracle.com/, 2008.

[23] phpmyadmin. http://www.phpmyadmin.net/, 2001.

[24] Postgresql. http://www.postgresql.org, 2008.

[25] Liam Quin. Extensible markup language (xml) - http://www.w3.org/xml/, 2008.

[26] Microsoft SQL Server. http://www.microsoft.com/sql/default.mspx, 2008.

[27] Chris Shiflett. *Essential PHP security*. O'Reilly, Essential PHP security, 2006.

[28] Oracle Text. http://www.orafaq.com/faqctx.htm, 2008.

[29] HP Open View. http://h20229.www2.hp.com/products/sdesk, 2008.

[30] Wiki. http://pt.wikipedia.org/wiki/wiki, 2008.

[31] Wikimatrix. http://www.wikimatrix.org/compare/dokuwiki+mediawiki, 2008.

[32] Wikipedia. http://pt.wikipedia.org/wiki/wikipedia, 2008.

[33] Nicholas C. Zakas, Jeremy McPeak, and Joe Fawcett. *Professional Ajax*. John Wiley and Sons, 2006.